APPLICATION OF PENTAGONAL AND TRIANGULAR FUZZY NUMBERS IN NEURAL NETWORK [BACK PROPAGATION ALGORITHM]

A. Virgin Raj and V. Nandhini

PG & Research Department of Mathematics, St. Joseph's College of Arts and Science (Autonomous), Cuddalore, Tamil Nadu, India.

A B S T R A C T   **RESEARCH ARTICLE**

In this Paper, We find out which quality of fertilizer is used to get an expected output, using the triangular fuzzy number and the pentagonal fuzzy number in Neural Network Back propagation algorithm.

Keywords:

Fuzzy number, Pentagonal fuzzy number, Triangular fuzzy number, activation function, back propagation neural network.

1. Introduction

In 1965, Zadeh Introduced the Fuzzy sets to represent and information possessing non-statistical certainties. However, the story of fuzzy logic started much earlier. The rotation of an infinite valued logic was "Fuzzy Set" where he described the mathematics of fuzzy set theory, and by extension fuzzy logic. Many operations were carried out using fuzzy numbers. Fuzzy neural networks are usually based on neural network architecture with fuzzification of inputs, outputs, weights, or rules that are applied using fuzzy system.

2. Preliminaries

Definition 2.1 (Fuzzy set). A fuzzy set is any set that allows its members to have different degree of membership called membership function, in the interval [0, 1]. A fuzzy set A is defined in the universal space X and denoted by an ordered set of pairs, the first element of which denotes the element and the second degree of membership i.e., $\bar{A} = \{ (x, \mu_{\bar{A}}(x)); x \in X \}$.

Definition 2.2 (Fuzzy number). A fuzzy number $\bar{A}$ is a fuzzy set on the real line R, must satisfy the following conditions

1. $\mu_{\bar{A}}(x_0)$ is the piecewise continuous
2. There exist at least one $x_0 \in R$ with $\mu_{\bar{A}}(x_0)=1$
3. $\bar{A}$ must be normal and convex.

Definition 2.3 (Triangular fuzzy number). Triangular fuzzy number is defined as $\bar{A} = \{a,b,c\}$ where all a,b,c are real numbers and its membership function is given below

$$\mu_{\bar{A}}(x_0) = \begin{cases} \frac{(x-a)}{(b-a)}, & for\ a \leq x \leq b \\ \frac{(c-x)}{(c-b)}, & for\ b \leq x \leq c \\ 0, & otherwise. \end{cases}$$

Definition 2.4 (Pentagonal fuzzy number). A pentagonal fuzzy number of a fuzzy set P is defined as P={a,b,c,d,e} and its membership functions is given by

$$\mu_P(x) = \begin{cases} 0, & for\ x<a \\ \frac{(x-a)}{(b-a)} & for\ a \leq x \leq b \\ \frac{(x-b)}{(c-a)} & for\ b \leq x \leq c \\ 1, & for\ x=c \\ \frac{(d-x)}{(d-c)} & for\ c \leq x \leq d \\ \frac{(e-x)}{(e-d)} & for\ d \leq x \leq e \\ 0, & for\ x > e \end{cases}$$
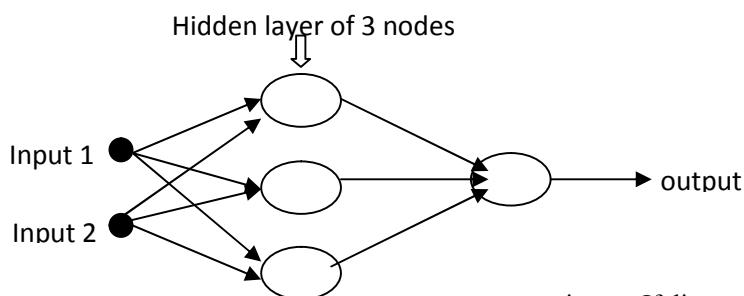
3. Neural Network

An artificial neural network is an information processing system that has certain performance characteristics in common with biological neural network. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology, based on the assumptions that:

i)   Information processing occurs at many simple elements called neurons.

ii) Signals are passed between neurons over connection links.

iii) Each connection link has an associated weight, which multiplies the signal transmitted.

iv) Each neuron applies an activation function (Usually nonlinear) to its net input (Sum of Weighted Input Signals) to determine its Output Signals.

The following figure illustrates a fully connected two input, Single-output, feed-forward, multilayer network with a single hidden layer consisting of three nodes.

Hidden layer of 3 nodes

4. Activation Function

According to Faqs. Org [2010] activation functions are needed for hidden layer of the NN to introduce non-linearity. Without them Neural Network would be same as plain perceptions. If linear function were used, Neural Network would not be as powerful as they are. Activation function can be linear, threshold or sigmoid function. Sigmoid activation function is usually used for hidden layer. The following figure will be used to illustrate activation function.
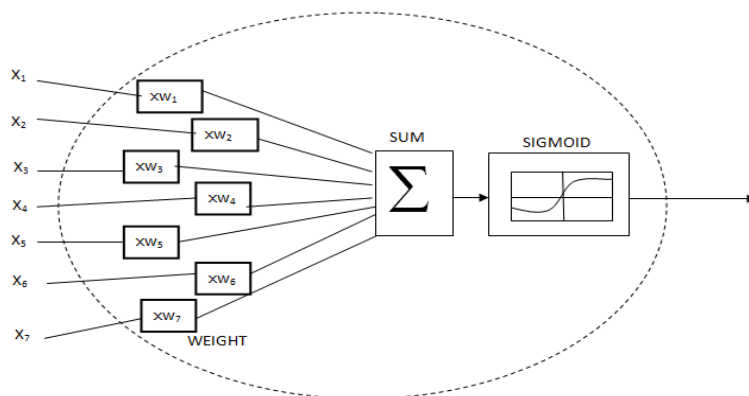
SUM is collection of the output nodes from hidden layer that have been multiplied by connection weights, added to get single number output through sigmoid function (activation function).

5. Back propagation Neural Network

Back propagation is a common method for training a neural network. The goal of back propagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs. Our goal with back propagation is to update each of the weights in the network so that they cause the actual output to be closer the target output, thereby minimizing the error for each output neuron and the network as a whole. In feed forward the network move towards only, but in back propagation we move backward also. In this section we use back propagation algorithm.

6. Algorithm

Step 1: Initialize the expected Output

Step 2: List the values for the problem which is in the form of pentagonal fuzzy number for input layer and triangular fuzzy number for hidden layer.

Step 3: Convert the pentagonal fuzzy number and triangular fuzzy number to its membership function.

Step 4: Set the pentagonal fuzzy number and triangular fuzzy number as the weights of the input layer and hidden layer respectively.

Step 5: Assume the input values 0 and 1

Step 6: For a Hidden Unit (F_j , j=1,2,3) Calculate weighted input signal, $F\_in_j = v_{0j} + \sum x_i\, v_{ji}$

Step 7: To calculate output signal, Apply activation function(Here we use sigmoid function) $Fj = f(F\_in_j)$

Step 8: For a output unit (O_k , k=1,2,3), Calculate weighted input signal,

$$O\_in_k = w_{0k} + \sum_{j=1}^{3} F_j\, w_{kj}$$

Step 9: To calculate output signal, apply activation function $O_k = f(O\_in_k)$

If the calculated output signal is not equal to the expected output we go the next step.

To find Error:

Step 10: For output unit O_k (k=1,2,3)

Error information term $\delta_k = (t_k - o_k) f'(O\_in_k)$

Weight correction term $\Delta w_{jk} = \alpha \delta_k\, Fj$

Bias correction term $\Delta w_{0k} = \alpha \delta_k$

Where α is a learning rate lies between 0.5 and 1.5

Step 11: For Input unit F_j (j=1,2,3)

Delta input $\delta\_in_j = \sum \limits^{3} w_{jk}$

Error information term $\delta_j = \sum\limits_{k=1} \delta_j$ f $'$ (F_in$_j$)

Weight correction term $\quad \Delta v_{ji} = \alpha\delta_j \, x_i$

Bias correction term $v_{0j} = \alpha\delta_j$

Step 12: Now update the weights and biases using the following formula

For a output unit (O_k , j,k=1,2,3)

$\qquad w_{kj}(new) = w_{kj}$ (old) + Δw_{kj}

For a Hidden unit (Fj, j=1,2,3)

$\qquad V_{ji}(new) = v_{ji}(old) + \Delta v_{ji}$ (i=1,2,3,4,5)

Step 13: Repeat the step from 5 to 8, Stop the process when the calculated output is equal to the expected output, and otherwise repeat the process.

7. Numerical Example

Suppose F1, F2, F3 are three fertilizers. Let their possible attributes to the above fertilizer v={a,b,c,d,e} as Universal set, where a,b,c,d,e represent components of fertilizers.   Let w={f,g,h} be the possible attributes of the lands , where f, g, h represent quality of fertilizers. Estimate the weights of fertilizers for the expected output (0.80, 0.85, 0.99).

Step 1: Assume the Expected output (0.80, 0.85, 0.99)

Step 2

$V_1 = (1, 2, 3, 4, 5)$, $V_2 = (2, 3, 4, 5, 6)$,

$V_3 = (3, 4, 5, 6, 7)$

$W_1 = (1, 2, 3)$, $\qquad W_2 = (2, 3, 4)$,

$W_3 = (6, 7, 8)$

Step 3

$V_1 = (0.1, 0.2, 0.3, 0.4, 0.5)$,

$V_2 = (0.2, 0.3, 0.4, 0.5, 0.6)$,

$V_3 = (0.3, 0.4, 0.5, 0.6, 0.7)$

$W_1 = (0.1, 0.2, 0.3)$, $\quad W_2 = (0.2, 0.3, 0.4)$, $\quad W_3 = (0.6, 0.7, 08)$

Step 4

$v_{11} = 0.1$, $v_{12} = 0.2$, $v_{13} = 0.3$, $v_{14} = 0.4$,

$v_{15} = 0.5$

$v_{21} = 0.2$, $v_{22} = 0.3$, $v_{23} = 0.4$, $v_{24} = 0.5$,

$v_{25} = 0.6$

$v_{31} = 0.3$, $v_{32} = 0.4$, $v_{33} = 0.5$, $v_{34} = 0.6$,

$v_{35} = 0.7$

$w_{11} = 0.1$, $w_{12} = 0.2$, $w_{13} = 0.3$

$w_{21} = 0.2$, $w_{22} = 0.3$, $w_{23} = 0.4$

$w_{31} = 0.6$, $w_{32} = 0.7$, $w_{33} = 0.8$

Let $v_{0i} = 0.20$ and $w_{0j} = 0.35$ i,j = 1,2,3

Step 5

Input signal $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 0, 1)$

Step 6

$F\_in_1 = v_{01} + x_1v_{11} + x_2 v_{12} + x_3v_{13} + x_4 v_{14} + x_5v_{15}$

$= .20 + (1)(0.1) + (1)(0.2) + (0)(0.3) + (0)(0.4) + (1)(0.5)$

$F\_in_1 \quad = 1$

$F\_in_2 = v_{02} + x_1v_{21} + x_2 v_{22} + x_3 v_{23} + x_4 v_{24} + x_5 v_{25}$

$= .20 + (1)(0.2) + (1)(0.3) + (0)(0.4) + (0)(0.5) + (1)(0.6)$

$F\_in_2 \quad = 1.3$

$F\_in_3 = v_{03} + x_1v_{31} + x_2 v_{32} + x_3 v_{33} + x_4 v_{34} + x_5 v_{35}$

$= .20 + (1)(0.3) + (1)(0.4) + (0)(0.5) + (0)(0.6) + (1)(0.7)$

$F\_in_3 \quad = 1.6$

Step 7

$F_1 = f(F\_in_1)$

$\qquad = \dfrac{1}{1+\exp(-F\_in1)}$

$\qquad = \dfrac{1}{1+\exp(-1)}$

$\qquad = \dfrac{1}{1+0.3678\ 794}$

$\qquad = \dfrac{1}{1.3678\ 794}$

$\qquad = 0.7310586$

$F_1 = 0.731$

$F_2 = f(F\_in_2)$

$\qquad = \dfrac{1}{1+\exp(-F\_in2)}$

$\qquad = \dfrac{1}{1+\exp(-1.3)}$

$\qquad F_2 = 0.795$

$F_3 = f(F\_in_3)$

$\qquad = \dfrac{1}{1+\exp(-F\_in3)}$

$\qquad = \dfrac{1}{1+\exp(-1.6)}$

$\qquad F_3 = 0.832$

Step 8

$O\_in_k = w_{0k} + \sum \limits_{j=1} F_j \, w_{kj}$

$O\_in_1 = w_{01} + F_1w_{11} + F_2w_{12} + F_3w_{13}$

$O\_in_1 = 0.35 + (0.73)(0.1) + (0.79)(0.2) + (0.83)(0.3)$

$O\_in_1 = 0.83$

$O\_in_2 = w_{02}+ F_1w_{21} +F_2w_{22} +F_3w_{23}$

$O\_in_2 = 1.065$

$O\_in_3 = w_{03}+ F_1w_{31} +F_2w_{32} +F_3w_{33}$

$O\_in_3 = 2.005$

Step 9

$O_k = f(O\_in_k)$

$O_1= f(O\_in_1)$

$$= \frac{1}{1+\exp(-0.8\ 3)}$$

$O_1=0.69$

$O_2 = f(O\_in_2) = 0.74$

$O_3 = f(O\_in_3) = 0.88$

$O_1 = 0.69$, $O_2 = 0.74$, $O_3 = 0.88$, which is not a expected output.

Step 10

For Output unit O_k (k = 1,2,3)

Error information term, $\delta_k = (t_k - O_k)\ f\ '(O\_in_k) = (t_k- O_k)f(O\_in_k)(1-f(O\_in_k))$

$\delta_1 = (t_1-O_1)f(O\_in_1)(1-f(O\_in_1))$

$\quad =(0.80-0.69)(0.69)(1-0.69)$

$\delta_1 =0.023529$

$\delta_2 = (t_2-O_2)f(O\_in_2)(1-f(O\_in_2))$

$\quad =(0.85-0.74)(0.74)(1-0.74)$

$\delta_2 =0.021164$

$\delta_3 = (t_3-O_3)f(O\_in_3)(1-f(O\_in_3))$

$\quad =(0.99-0.88)(0.88)(1-0.88)$

$\delta_3 =0.011616$

Let $\alpha = 0.8$

Weight correction terms:

$\Delta w_{11} = 0.0137 \quad \Delta w_{12} = 0.0124 \quad \Delta w_{13} = 0.0068$

$\Delta w_{21} = 0.0149 \quad \Delta w_{22} = 0.0134 \quad \Delta w_{23} = 0.0073$

$\Delta w_{31} = 0.0156 \quad \Delta w_{32} = 0.0141 \quad \Delta w_{33} = 0.0077$

Bias correction term:

$\Delta w_{01} = \alpha\delta_1 = (0.8)\ (0.023529) =0.0188$

$\Delta w_{02} = \alpha\delta_2 = (0.8)\ (0.021164) =0.0169$

$\Delta w_{03} = \alpha\delta_3 = (0.8)\ (0.011616) =0.0093$

Step 11

For Input Unit Fj (j=1,23)

Delta Input $\delta\_in_j = \sum_{k=1}^{3} w$

$\delta\_in_1 = \delta_1w_{11} + \delta_2w_{12}+\delta_3w_{1:}$

$\quad = (0.023529)(0.1) + (0.021164)(0.2) + (0.011616)(0.3)$

$\delta\_in_1=0.010071$

$\delta\_in_2 =0.015700$

$\delta\_in_3=0.038225$

Error information term $\delta_j = \delta\_in_j\ f\ '(F\_in_j) = \delta\_in_j\ f\ (F\_in_j)(1- f\ (F\_in_j))$

$\delta_1 = \delta\_in_1\ f\ (F\_in_1)\ (1- f\ (F\_in_1))$

$\quad = (0.010071)(0.73)(1-0.73) = 0.001985$

The errors are given by,

$\delta_1 = 0.001985$

$\delta_2 = 0.002605$

$\delta_3 = 0.005394$

Weight correction terms :

$\Delta v_{11} = 0.001588; \quad \Delta v_{12} = 0.001588; \quad \Delta v_{13} = 0; \quad \Delta v_{14} = 0; \quad \Delta v_{15} = 0.001588$

$\Delta v_{21} = 0.002084; \quad \Delta v_{22} = 0.002084; \quad \Delta v_{23} = 0; \quad \Delta v_{24} = 0; \quad \Delta v_{25} = 0.002084$

$\Delta v_{31} = 0.004315; \quad \Delta v_{32} = 0.004315; \quad \Delta v_{33} = 0; \quad \Delta v_{34} = 0; \quad \Delta v_{35} = 0.004315$

$\Delta v_{01} = \alpha\delta_1 =(0.8)\ (0.001985) = 0.001588$

$\Delta v_{02} = \alpha\delta_2 =(0.8)\ (0.002605) = 0.002084$

$\Delta v_{03} = \alpha\delta_3 =(0.8)\ (0.005394) = 0.004315$

Step 12

Updated weights and biases

For output unit (O_k, j,k = 1,23), $w_{kj}(new) = w_{kj}(old) + \Delta w_{kj}$

w_{11} (new) = 0.1137; $\quad w_{12}$ (new) = 0.2124; $\quad w_{13}$ (new) = 0.3068

w_{21} (new) = 0.2149; $\quad w_{22}$ (new) = 0.3134; $\quad w_{23}$ (new) = 0.4073

w_{31} (new) = 0.6156; $\quad w_{32}$ (new) = 0.7141; $\quad w_{33}$ (new) = 0.8077

w_{01} (new) = 0.3688; $\quad w_{02}$ (new) = 0.3669; $\quad w_{03}$ (new) = 0.3593

For input unit (F_j, i = 1, 2, 3, 4, 5) , v_{ji} (new) = v_{ji} (old) + Δv_{ji}

$v_{11}(new)=0.10159; \quad v_{12}(new)=0.20159; \quad v_{13}(new)=0.3;$ $v_{14}(new)=0.4; v_{15}(new)=0.50159$

$v_{21}(new)=0.20208; \quad v_{22}(new)=0.30208; \quad v_{23}(new)=0.4;$ $v_{24}(new)=0.5; v_{25}(new)=0.60208$

v_{31}(new)=0.30431; v_{32}(new)=0.40431; v_{33}(new)=0.5; v_{34}(new)=0.6; v_{35}(new)=0.70431

v_{01} (new)=0.201588; v_{02}(new)=0.202084; v_{03}(new)=0.204315

Updated Weights:

v_{11} = 0.10159; v_{12} = 0.20159; v_{13} = 0.3; v_{14} = 0.4; v_{15} = 0.50159

v_{21} = 0.20208; v_{22} = 0.30208; v_{23} = 0.4; v_{24} = 0.5; v_{25} = 0.60208

v_{31} =0.30431; v_{32} = 0.40431; v_{33} = 0.5; v_{34} = 0.6; v_{35} = 0.70431

v_{01} = 0.201588; v_{02} = 0.202084; v_{03} = 0.204315

w_{11} = 0.1137; w_{12} = 0.2124; w_{13} = 0.3068

w_{21}= 0.2149; w_{22} = 0.3134; w_{23} = 0.4073

w_{31} = 0.6156; w_{32}= 0.7141; w_{33} = 0.8077

w_{01} = 0.3688; w_{02} = 0.3669; w_{03} = 0.3593

Step 13

Using the updated weights, we get outputs (0.71, 0.74, 0.89), which is not equal to the expected output we proceed the steps rectifying the errors and till we reach the expected value. After repeating the process so many times we get, updated weights,

v_{11} = 0.16159; v_{12} = 0.26159; v_{13} = 0.3; v_{14} = 0.4; v_{15} = 0.56159

v_{21} = 0.27208; v_{22} = 0.37208; v_{23} = 0.4; v_{24} = 0.5; v_{25} = 0.67208

v_{31} =0.99431; v_{32} = 0.99431; v_{33} = 0.5; v_{34} = 0.6; v_{35} = 0.99431

v_{01} = 0.271588; v_{02} = 0.312084; v_{03} = 0.454315

w_{11} = 0.2837; w_{12} = 0.3424; w_{13} = 0.4268

w_{21}= 0.4749; w_{22} = 0.4413; w_{23} = 0.5273

w_{31} = 0.9556; w_{32}= 0.9841; w_{33} = 0.9977

w_{01} = 0.5488; w_{02} = 0.5269; w_{03} = 0.9593

Using these weights we get expected output (0.80, 0.85, 0.99).

Conclusion

So far we use the pentagonal fuzzy numbers and triangular fuzzy numbers in Neural Network Back propagation algorithm to rectify errors in weights and using these updated weights we find out expected output (0.80, 0.85, 0.99).

Reference

[1]   H.K. Baruah, Towards forming field of fuzzy sets, International Journal of energy, Information and communications, 2(1)(2011),16-20

[2]   A. Carling, Introducing Neural Networks, Sigma Press-Wilmslow, United Kingdom, (1992)

[3]   G. Facchientti and N. Pacchiaroti, Evaluations of fuzzy quantities, fuzzy sets and systems, 157(2 006), 892-903

[4]   Laurance V. Fausett, Fundamentals of Neural network Architecture Algorithm and applications, Publisher – Printice Hall, 1994.

[5]   D. Gruape, Principles of artificial neural networks (2nd edition), World Scientific Publishing, (2007).

[6]   George J.Klie and Boyan, Fuzzy sets and Fuzzy Logic theory and applications, Prentice-Hall, (1995).

[7]   M. M. Gupta and D.H. Rao, On the Principles of fuzzy Neural Networks, Fuzzy sets and systems 61(1994) 1-18.

[8]   L.A Ladeh, Fuzzy sets, Information and control, (1965), 338-353.

[9]   Raul Rojas, Neural Networks, Springer-Verlag, Berlin 1996.

[10]   Robert Fuller, Neural Fuzzy System.

[11]   Thomas Feuring , Learning in fuzzy neural network.

[12]   Vikaskumar and Dr. J. S. Sohal, Fuzzy Neural Networks, IJCSMS International Journal of Computer science and Management Studies, Vol.12, Issue 03 ISSN: 2231-5268.

[13]   A. Virginraj and S. Karthik, An Application of Pentagonal fuzzy number in neural network, International Journal of mathematics and its applications, Volume 4, Issue 412016, ISSN 2347-1557.

* * * * * * *